

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A computer-implemented method for efficiently parsing received data files, comprising:
 - receiving, by a virtual browser, a data file;
 - retrieving, by the virtual browser, a stored version of the data file and a syntax tree comprising nodes and tokens representing data within the data file, the tree including at least one static node;
 - comparing, by a comparison engine in communication with the virtual browser, the stored version of the data file with the received data file to identify non-matching content in the received data file;
 - parsing, by a parsing engine of the virtual browser, only the non-matching content of the received data file to form at least one subtree comprising nodes and tokens representing the non-matching content of the received data file;
 - replacing, by the virtual browser, at least one static node of the syntax tree with a token; and
 - creating, by the virtual browser, a mapping from each token to one of the subtrees.
2. (Canceled)
3. (Canceled)
4. (Previously Presented) The computer-implemented method of claim 1 wherein the data file is a web page.
5. (Previously Presented) The computer-implemented method of claim 1 wherein the data file is an HTML file.
6. (Currently Amended) A method for efficiently parsing web pages, comprising:
 - receiving, by a virtual browser, a first HTML page;
 - retrieving, by the virtual browser, a cached version of the HTML page and a syntax tree comprising nodes and tokens representing data within the first HTML page, the tree including at least one static node;

comparing, by a comparison engine in communication with the virtual browser, the cached version of the HTML page with the received HTML page to identify non-matching content in the received HTML page;

parsing, by a parsing engine of the virtual browser, only the non-matching content in the received HTML page to form at least one subtree comprising nodes and tokens representing the non-matching content of the received data file;

replacing, by the virtual browser, at least one static node of the syntax tree with a token;
and

creating, by the virtual browser, a mapping from each token to one of the subtrees.

7. (Canceled)

8. (Currently Amended) A method for efficiently parsing HTML pages,
comprising:

receiving, by a virtual browser, a first HTML page;

responsive to a determination that a cached version of the HTML page exists:

retrieving, by the virtual browser from a cache, the cached version of the HTML page and a first syntax tree comprising nodes and tokens representing data within the first HTML page, the first tree including at least one static node;

comparing, by a comparison engine in communication with the virtual browser, the cached version of the first HTML page with the received HTML page to identify non-matching content in the received HTML page;

parsing, by a parsing engine of the virtual browser, only the non-matching content to form a subtree;

creating, by the virtual browser, a mapping from a token of the first tree to the subtree;

responsive to a determination that the cached version of the HTML page does not exist:

parsing, by the parsing engine of the virtual browser, the received HTML page to form a second syntax tree comprising nodes and tokens representing the non-matching content of the received data file, the second tree containing at least one static node; and

storing the second tree and the received HTML page in the cache.

9. (Currently Amended) A method for providing derivative services comprising:
- receiving, by a virtual browser, a first HTML page;
 - constructing a syntax tree comprising nodes and tokens representing data within the received HTML page, the tree comprising a plurality of nodes;
 - ~~determining that~~ at least one node of the tree contains static content;
 - determining, by a comparison engine in communication with the virtual browser, that at least one node of the tree contains dynamic content;
 - replacing, by the virtual browser, the nodes of the tree containing dynamic content with tokens;
 - parsing, by a parsing engine of the virtual browser, only the dynamic content to form subtrees representing the dynamic content of the received data file; and
 - mapping, by the virtual browser, the tokens to the subtrees.
10. (Currently Amended) A computer-implemented method of providing derivative services from a plurality of primary service providers, comprising:
- receiving, by a virtual browser, a request for derivative services content from a client node of a customer;
 - retrieving, by the virtual browser, data from a plurality of primary service providers on behalf of the customer; by:
 - identifying, by the virtual browser, static content that has been previously retrieved from the primary service providers and stored in a cache, and corresponding syntax trees comprising nodes and tokens representing data within the static content that have also been stored in the cache;
 - identifying, by a comparison engine in communication with the virtual browser, dynamic content that differs from the previously retrieved content;
 - parsing, by a parsing engine of the virtual browser, only the dynamic content to form subtrees representing the dynamic content of the received data file;
 - adding, by the virtual browser, tokens to the syntax trees;
 - mapping, by the virtual browser, the tokens to the subtrees;
 - creating, by the virtual browser, at least one content page comprising the retrieved data;
- and

providing, by the virtual browser, the created pages to the client node of the customer.

11. (Currently Amended) A method for efficiently parsing received data files, comprising:

receiving, by a virtual browser, a first data file;

retrieving a stored syntax tree from a cache, the stored syntax tree comprising nodes and tokens, ~~the stored syntax tree~~ representing data within the first data file and containing at least one static node and at least one token;

retrieving, by the virtual browser, a second data file from the cache, the second data file associated with the first data file;

identifying, by a comparison engine in communication with the virtual browser, non-matching content present only in the first data file;

parsing, by a parsing engine of the virtual browser, only the non-matching content of the first data file to form at least one subtree comprising nodes and tokens representing the non-matching content of the received data file; and

mapping, by the virtual browser, at least one of the tokens to at least one of the subtrees.

12. (Currently Amended) The method of claim 11, further comprising:

responsive to identifying non-matching content present only in the first file:

adding, by the virtual browser, at least one new token to the syntax tree.

13. (Currently Amended) A system for efficiently parsing input data from a plurality of content servers,

comprising:

~~at least one~~ a virtual browser for retrieving content from ~~primary~~ content servers;

an identification engine, in communication ~~very coupled to~~ with the virtual browser for identifying retrieved content;

a cache, in communication ~~very coupled to~~ with the virtual browser ~~and the parsing engine~~, for storing retrieved content and syntax trees comprising nodes and tokens representing data within the retrieved content;

a comparison engine in communication ~~coupled to~~ with the virtual browser, for comparing retrieved content with stored content to identify ~~differing non-matching~~ content not stored in the cache;

a parsing engine of the virtual browser, ~~communicatively coupled to the virtual browser,~~
for parsing only the non-matching content identified by the comparison engine ~~as differing~~
~~content,~~ and forming subtrees comprising nodes and tokens representing the non-matching
~~differing~~ content of the received data file and creating a mapping from new tokens to formed
subtrees; and

~~a content server, coupled to the virtual browser.~~

14. (Canceled)

15. (Currently Amended) ~~An computer program product~~ intermediary for efficiently parsing
received data files transmitted between a client and a server, the intermediary comprising:

~~computer program product stored on a computer readable medium and including~~
~~instructions for causing a computer to carry out the steps of:~~
~~receiving a data file;~~

a cache ~~retrieving a stored~~ a version of the a data file received from a server and a
syntax tree comprising nodes and tokens representing data within the data file, the tree including
at least one static node;

a comparison engine comparing the stored version of the data file with the received data
file to identify non-matching content in the received data file; and

a virtual browser in communication with the comparison engine, retrieving the stored
version of the data file and the syntax tree from the cache, parsing only the non-matching
content of the received data file to form ~~at~~ at least one subtree comprising nodes and tokens
representing the non-matching content of the received data file, ~~;~~ replacing at least one static
node of the syntax tree with a token, ~~;~~ and creating a mapping from each token to one of the
subtrees.